WHAT IS CLAIMED IS:

1. A floating point operand data structure used in floating point computations and processing within a processing device, comprising:

a first portion of the data structure having floating point operand data; and

a second portion of the data structure having embedded status information associated with at least one status condition of the floating point operand data.

2. The floating point operand data structure of claim 1, wherein the at least one status condition is determined from the embedded status information without regard to memory storage external to the data structure.

3. The floating point operand data structure of claim 2, wherein the memory storage external to the data structure is a floating point status register.

4. The enhanced floating point operand data structure of claim 3, wherein the outcome of a conditional floating point instruction is based on the embedded status information without regard to contents of the floating point status register.

5. The floating point operand data structure of claim 1, wherein the second portion of the data structure comprises at least one bit that is indicative of the at least one status condition from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

6. The floating point operand data structure of claim 5, wherein the overflow status represents one in a group of a +OV status and a -OV status.

7. The floating point operand data structure of claim 6, wherein the overflow status is represented as a predetermined non-infinity numerical value.

8. The floating point operand data structure of claim 5, wherein the underflow status represents one in a group of a +UV status and a -UV status.

9. The floating point operand data structure of claim 5, wherein the underflow status is represented as a predetermined non-zero numerical value.

10. The floating point operand data structure of claim 5, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

11. The floating point operand data structure of claim 10, wherein the second portion of the data structure comprises a plurality of bits indicative of a predetermined type of operand condition resulting in the NaN status.

12. The floating point operand data structure of claim 10, wherein addition, multiplication, maximum and minimum floating point operations on the data structure are commutative.

13. The floating point operand data structure of claim 5, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.

14. The floating point operand data structure of claim 1, wherein the second portion of the data structure comprises a plurality of bits indicative of a predetermined type of operand condition resulting in the infinity status.

15. The floating point operand data structure of claim 1, wherein the at least one status condition is associated with at least one floating point operation that generated the enhanced floating point operand data structure.

16. A floating point operand data structure used by a processing device when performing floating point operations, the data structure comprising:

a first data field having sign information associated with the floating point operand;

a second data field having exponent information associated with the floating point operand; and

a third data field having fractional information associated with the floating point operand, wherein at least one of the first data field, the second data field and the third data field further includes embedded status information associated with at least one operand status condition.

17. The floating point operand data structure of claim 16, wherein the at least one operand status condition is determined from the embedded status information without regard to a floating point status register that is separate from an operand memory storage device for maintaining the floating point operand data structure.

18. The floating point operand data structure of claim 17, wherein the outcome of a conditional floating point instruction is based on the embedded status information without regard to contents of the floating point status register.

19. The floating point operand data structure of claim 16, wherein the embedded status information comprises at least one bit that is indicative of the at least one operand status condition from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

20. The floating point operand data structure of claim 16, wherein the at least one operand status condition is indicative of at least one floating point operation that generated the floating point operand data structure.

21. A floating point system associated with a processing device for performing at least one floating point operation on a floating point operand, the system comprising:

an operand memory storage device for maintaining the floating point operand;

a control unit in communication with the operand memory storage device, the control unit receiving at least one floating point instruction associated with the at least one floating point operation and generating at least one control signal related to the at least one floating point operation; and

a first functional processing unit in communication with the operand memory storage device and the control unit, the first functional processing unit capable processing the floating point operand and storing status information within the processed floating point operand.

22. The floating point system of claim 21, wherein the status information comprises at least one bit that is indicative of an operand status condition from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

23. The floating point system of claim 21, wherein the first functional processing unit is capable of embedding the status information related to the processed floating point operand within predetermined fields of the processed floating point operand.

24. The floating point system of claim 21, wherein the first functional processing unit is capable of providing the processed floating point operand to the

operand memory storage device without storing the status information to a separate status memory device.

25. The floating point system of claim 24, wherein the control unit is operative to condition the outcome of the floating point instruction based upon the status information within the processed floating point operand without accessing the separate status memory device.

26. The floating point system of claim 21 further comprising a second functional processing unit in communication with the memory storage device and the control unit, the second functional processing unit being capable of processing a second floating point operand and storing status information related to the second floating point operand while the status information related to the first floating point operand is preserved.

27. A floating point processing system for performing at least one floating point operation on a floating point operand, the system comprising:

an operand memory register for maintaining the floating point operand; and

a functional processing unit in communication with the operand memory register, the functional processing unit being operative to:

receive the floating point operand from the operand memory register,

process the floating point operand to determine status information related to the processed floating point operand, and

embed the status information within the processed floating point operand.

28. The floating point system of claim 27, wherein the status information comprises at least one bit that is indicative of an operand status condition from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

29. The floating point system of claim 27, wherein the functional processing unit is further operable to embed the status information within at least one predetermined field of the processed floating point operand.

30. The floating point system of claim 27, wherein the functional processing unit is capable of storing the processed floating point operand in the operand memory register without storing the status information in a floating point status register separate from the operand memory register.

31. The floating point system of claim 27 further comprising a control unit in communication with the operand memory register and the functional processing unit, the control unit being operative to condition the outcome of the floating point instruction based only upon the status information within the processed floating point operand.

32. The floating point system of claim 31 further comprising an additional functional processing unit in communication with the operand memory register and the control unit, the additional functional processing unit being capable of concurrently processing an additional floating point operand and storing status information related to the additional floating point operand within the additional floating point operand while the status information related to the other floating point operand is preserved within the other floating point operand.

33. A method of encoding a floating point operand with status information without maintaining the status information in a floating point status register, comprising:

determining a status condition of the floating point operand as part of processing the floating point operand in association with a floating point operation; and

representing an updated status condition of the floating point operand within the floating point operand.

34. The method of claim 33, wherein the determining step further comprises identifying the status condition and the updated status condition from only embedded status information within the floating point operand.

35. The method of claim 33, wherein the status condition and the updated status condition are from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

36. The method of claim 33, wherein the representing step further comprises embedding updated status information within the floating point operand after execution of the floating point operation, the updated status information representing the updated status condition.

37. The method of claim 33, wherein the updated status condition is indicative of a previous floating point operation that resulted in the floating point operand.

38. The method of claim 33 further comprising conditioning a subsequent floating point operation based only upon the updated status information within the floating point operand.

39. The method of claim 33 further comprising processing an additional floating point operand and representing updated status information related to the additional floating point operand within the additional floating point operand while the updated status information related to the other floating point operand is preserved.

40. A method of encoding a floating point operand with status information related to a status of the floating point operand without maintaining the status information in a floating point status register, comprising:

receiving a floating point instruction;

accessing a floating point operand to be processed as part of processing the floating point instruction;

decoding an initial status condition of the floating point operand from only status information embedded within the floating point operand; and

encoding a resulting status condition from execution of the floating point instruction on the floating point operand as updated status information within the floating point operand.

41. The method of claim 40 further comprising conditioning execution of a subsequent floating point instruction based only on the updated status information within the floating point operand.

42. The method of claim 40, wherein the initial status condition and resulting status condition are from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

43. The method of claim 40 further comprising processing an additional floating point operand and encoding an updated status information into the additional floating point operand while the updated status information related to the other floating point operand is preserved.

44. A computer-readable medium on which is stored a set of instructions for encoding a floating point operand with status information without maintaining the status information in a floating point status register, which when executed perform steps comprising:

determining a status condition of the floating point operand as part of processing the floating point operand in association with a floating point operation; and

representing an updated status condition of the floating point operand within the floating point operand.

45. The computer-readable medium of claim 44, wherein the determining step further comprises identifying the status condition and the updated status condition from only embedded status information within the floating point operand.

46. The computer-readable medium of claim 44, wherein the status condition and the updated status condition are from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

47. The computer-readable medium of claim 44, wherein the representing step further comprises embedding updated status information within the floating point operand after execution of the floating point operation, the updated status information representing the updated status condition.

48. The computer-readable medium of claim 44, wherein the updated status condition is indicative of a previous floating point operation that resulted in the floating point operand.

49. The computer-readable medium of claim 44 further comprising conditioning a subsequent floating point operation based only upon the updated status information within the floating point operand.

50. The computer-readable medium of claim 44 further comprising processing an additional floating point operand and representing updated status information related to the additional floating point operand within the additional floating point operand while the updated status information related to the other floating point operand is preserved.

51. A computer-readable medium on which is stored a set of instructions for encoding a floating point operand with status information related to a status of the floating point operand without maintaining the status information in a floating point status register, which when executed perform steps comprising:

receiving a floating point instruction;

accessing a floating point operand to be processed as part of processing the floating point instruction;

decoding an initial status condition of the floating point operand from only status information embedded within the floating point operand; and

encoding a resulting status condition from execution of the floating point instruction on the floating point operand as updated status information within the floating point operand.

52. The computer-readable medium of claim 51 further comprising conditioning execution of a subsequent floating point instruction based only on the updated status information within the floating point operand.

53. The computer-readable medium of claim 51, wherein the initial status condition and resulting status condition are from the group of an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

54. The computer-readable medium of claim 51 further comprising processing an additional floating point operand and encoding an updated status information into the additional floating point operand while the updated status information related to the other floating point operand is preserved.